Advanced Dashboard Widgets

Session 134

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Advanced Dashboard Widgets

Christian Wagner
Technical Writer

http://developer.apple.com/wwdc2005/

What We'll Cover

- Command Line Utilities and Scripts
- HTTP Requests and XML Feeds
- Widget Plug-ins
- Web Kit Plug-ins
- Other useful bits

From Yesterday

- The basic Widget
- Design tips
- Localization
- Preferences and Widget backs
- Widget events

A Note About Access Keys

AllowFileAccessOutsideOfWidget
AllowInternetPlugins
AllowJava
AllowNetworkAccess
AllowSystem
AllowFullAccess

Command Line Access

- Use `AllowSystem` Info.plist key
- Run command line utilities from a Widget

```
widget.system("command", handler)
```

- Two modes
  - Synchronous
  - Asynchronous

Synchronous widget.system

- Execution halts until command returns
- Used when no end handler provided

```
widget.system("/usr/bin/uptime", null)
```

Available Properties

outputString
errorString
status

Using Synchronous widget.system

```
var foo = widget.system(“/usr/bin/date”, null).outputString;
```

```
var foo = widget.system(“/usr/bin/date”, null);
var fooOutput = foo.outputString;
var fooStatus = foo.status;
```

Demo

Synchronous Command Line Access

http://developer.apple.com/wwdc2005/

Asynchronous widget.system

- Command executes concurrently
- Used when a handler set

```
var foo = widget.system(“/usr/bin/say ‘Hello’”, endHandler);
```

Asynchronous widget.system Handlers

Handlers

onreadoutput
onreaderror

```
var foo = widget.system(“./script.pl”, myEndHandler);
foo.onreadoutput = outputReceived;
...
function outputReceived(output)
{
      document.element.innerText = output;
}
```

Asynchronous widget.system Interaction

Methods

cancel()
write(string)
close()

```
var foo = widget.system(“./script.pl”, myEndHandler);
foo.write(“some string”);
foo.close();
```

Demo

Asynchronous Command Line Access

http://developer.apple.com/wwdc2005/

Network Resource Access

Matt Drance
DTS Engineer

http://developer.apple.com/wwdc2005/

Fetching Web-based Content

- Set `AllowNetworkAccess` Info.plist Key
- Use `XMLHttpRequest` object to fetch XML/HTML content
  - Instantiate in JavaScript
  - Set handlers
  - Open HTTP connection
  - Send request
  - Process response in handler function

Setting Up a Request

- Set a handler function
  - `onreadystatechange`
  - `onload`
- Set request type and URL using `open`
  - Pass `true` for asynchronous mode
- Set request headers (optional)
- Send the request

Setting Up a Request

```
var req;

function loadXML(url) {
    if (window.XMLHttpRequest) {
        req = new XMLHttpRequest();
        req.onreadystatechange = processReqChange;
        req.open("GET", url, true);
        req.send(null);
    }
}
```

Handling a Response

- Handler called
- Inspect server response
  - `getResponseHeader(“headerName”)`
  - `getAllResponseHeaders`
  - `requestText`
  - `requestXML`
- Inspect request properties
  - `onreadystatechange`: check `readyState, status`

Handling a Response

```
function processReqChange() {
    if (req.readyState == 4) {
        if (req.status == 200) {
      if (req.responseXML) {
        processDOMTree(req.responseXML);
      } else {
          alert("Problem retrieving XML data:\n"
              + req.statusText);
      }
    }
 }
```

responseXML vs. responseText

- `responseXML`: functional DOM object
- `responseHTML`: raw document source
  - Stored as a JavaScript string
  - `indexOf`, `substring`, `match`, etc.
  - Requires "text/xml" response MIME
    - Call `overrideMimeType` before sending request
  - Standard calls and properties
    - `getElementById`, `getElementsByTagName`, `firstChild`, etc

Demo

Network Resource Access

http://developer.apple.com/wwdc2005/

Dashboard Plug-Ins

http://developer.apple.com/wwdc2005/

Dashboard Plug-Ins

- What is a Widget Plug-in?
  - JavaScript access to *non-graphical* native services
  - Data retrieval/API exposure
    - Example: Address Book
- What is an Internet Plug-in?
  - View-based, *graphical* addition to web content
  - Native drawing, components
  - NSAPI or Web Kit Plug-ins
    - Example: Phone Book

Widget Plug-ins

- Use `Plugin` Info.plist Key
- Foundation Bundle
- Implement `initWithWebView:`
- Implement WebScripting protocol methods
  - `windowScriptObjectAvailable:`
  - `isSelectorExcludedFromWebScript:`
  - `isKeyExcludedFromWebScript:`

Method Implementations

- `initWithWebView:`
  - Required for all Widget Plug-ins
  - Plug-in object initialization
- `windowScriptObjectAvailable:`
  - Identifies your plug-in to JavaScript
  - Add a property to `window` object

```
- (void) windowScriptObjectAvailable:(WebScriptObject *)wso {
    [wso setValue:self forKey:”MyPlugin”];
}
```

Optional Method Implementations

- `webScriptNameForSelector:`
  - Desired JavaScript syntax for given methods
  - Returns an NSString
- `isSelectorExcludedFromWebScript:`
  - Allows / Bars JavaScript access to given methods
  - Returns `YES` (excluded) or `NO` (exposed)
  - Defaults to `YES`
- `isKeyExcludedFromWebScript:`
  - Defaults to `YES`
  - Use accessors and `isSelectorExcludedFromWebScript:`

Demo

Widget Plug-ins

http://developer.apple.com/wwdc2005/

Web Kit Plug-ins

- Use `AllowInternetPlugins` Info.plist Key
- Use `embed` tag to display
- Cocoa Bundle
- Implement WebScripting, WebPlugIn protocols
  - `plugInViewWithArguments:`
  - `objectForWebScript`
  - `webPlugInInitialize / webPlugInDestroy`
  - `webPlugInStart / webPlugInStop`
  - `webPlugInSetIsSelected`

Method Implementations

- `plugInViewWithArguments`:
  - Required by WebKitPlugIn protocol
  - Initialize your Plug-in view
  - Arguments dictionary contains embed attributes:

```
+ (id) plugInViewWithArguments:(NSDictionary *)args {
    NSDictionary *webArgs;
    webArgs = [args objectForKey:WebPlugInAttributesKey];
    return self;
}
```

Widget Considerations

- Conserve cycles in hidden Widgets
  - Implement `webPlugInStart, webPlugInStop`
  - Expose to JavaScript
    - `isSelectorExcludedFromWebScript:`
    - `webScriptNameForSelector:`
  - Tie into Widget events
    - Call `webPlugInStart` from `widget.onshow`
    - Call `webPlugInStop` from `widget.onhide`

Demo

Web Kit Plug-ins

http://developer.apple.com/wwdc2005/

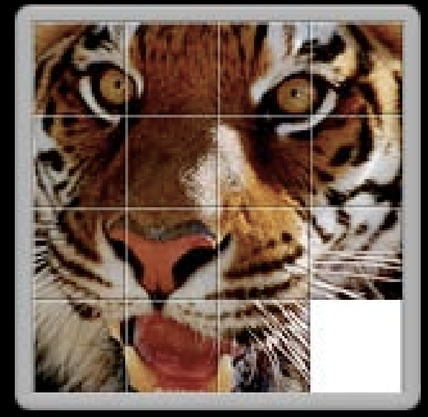
Other Useful Bits

Christian Wagner
Technical Writer

http://developer.apple.com/wwdc2005/

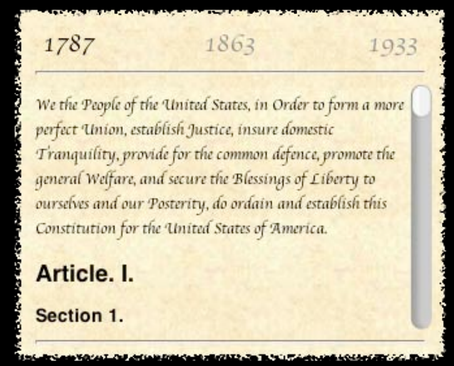
New Web Kit Technologies

- Canvas
- Search field
- Drag and drop
- Pasteboard operations



DHTML Effects

- Scroll bars and scrollers
- Stretching
- Live resizing
- Fading



What We've Covered

- Command Line Utilities and Scripts
- HTTP Requests and XML Feeds
- Widget Plug-ins
- Web Kit Plug-ins

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Who to Contact

Alan Samuel
Dashboard Evangelist
bluker1@apple.com

dashboard-dev@lists.apple.com
Widget Developer Community
http://lists.apple.com

Q&A

Christian Wagner
Technical Writer

Matt Drance
DTS Engineer

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference